Amendments to the Claims:

These claims will replace all prior versions, and listings, of claims in the application:

1-20. (Canceled)

21. (Previously presented) A program segment stored on a computer readable medium for cryptographically converting a digital input data block M into a digital output data block; said program segment comprising:

a program portion for merging a selected part M1 of said digital input data block M with a first digital key K1 to produce a data block B1 which non-linearly depends on said selected part M1 and said first digital key K1; and

a program portion for deriving said digital output block from said data block B1 and the remaining part of the digital input data block M, wherein said merging step is performed by executing a non-linear function g for non-linearly merging said selected part M1 and said first key K1 in a single step.

22. (Previously presented) A program segment as claimed in claim 21, comprising:

a program portion for splitting said digital input block into said selected part M1 and a second part M2 before executing said program portion for merging;

a program portion for executing a non-linear function g^{-1} to merge said second block M2 with a second key K2 in one step, producing a data block B2 as output; said non-linear function g^{-1} being the inverse of said non-linear function g; and

a program portion for forming combined data from data in said data block B1 and in said data block B2; said digital output block being derived from said combined data.

23. (Currently amended) A program segment as claimed in claim 21, wherein said program portion for merging comprises:

a program portion for splitting said selected part M1 in a first plurality n of sub-blocks $m_0,..,m_{n-1}$ of ~~substantially~~ equal length;

a program portion for splitting said first key K1 in said first plurality n of sub-keys $k_0,..,k_{n-1}$, ~~substantially~~ having equal length, the sub-key k_i corresponding to the sub-block m_i, for i = 0 to n-1;

a program portion for separately processing each of said sub-blocks m_i by executing for each of said sub-blocks m_i a same non-linear function h for non-linearly merging a sub-block b_i derived from said sub-block m_i with said corresponding sub-key k_i in one, sequentially inseparable step and producing said first plurality of output sub-blocks $h(b_i, k_i)$; and

a program portion for combining sub-blocks t_i derived from said first plurality of said output sub-blocks $h(b_i, k_i)$ to form said data block B1.

24. (Currently amended) A program segment as claimed in claim 22, wherein said program portion for executing said non-linear function g^{-1} comprises:

a program portion for splitting said second part M2 in said first plurality n of sub-blocks $m_n,..,m_{2n-1}$, ~~substantially~~ having equal length;

a program portion for splitting said key K2 in said first plurality n of sub-keys $k_n,..,k_{2n-1}$, ~~substantially~~ having equal length, the sub-key k_i corresponding to the sub-block m_i, for $i = n$ to $2n-1$;

a program portion for executing for each of said sub-blocks m_i a same non-linear function h^{-1} for non-linearly merging a sub-block b_i derived from said sub-block m_i with said corresponding sub-key k_i and producing said first plurality of an output sub-block $h^{-1}(b_i, k_i)$; said function h^{-1} being the inverse of said function h; and

a program portion for combining sub-blocks t_i derived from said first plurality of output sub-blocks $h^{-1}(b_i, k_i)$ to form said data block B2.

25. (Currently amended) A program segment as claimed in claim 23, wherein said sub-block b_i is derived from said sub-block m_i by bit-wise adding a constant p_i to said sub-block m_i, said constant p_i ~~substantially~~ having equal length as said sub-block m_i.

26. (Previously presented) A program segment as claimed in claim 23, characterised in that said function $h(b_i,k_i)$ is defined by:

$h(b_i,k_i) = (b_i.k_i)^{-1}$, if $b_i \neq 0$, $k_i \neq 0$, and $b_i \neq k_i$

$h(b_i,k_i) = (k_i)^{-2}$, if $b_i = 0$

$h(b_i,k_i) = (b_i)^{-2}$, if $k_i = 0$

$h(b_i,k_i) = 0$, if $b_i = k_i$,

where the multiplication and inverse operations are predetermined Galois Field multiplication and inverse operations.

27. (Currently amended) A program segment as claimed in claim 26, wherein deriving said sub-blocks t_i from said output sub-blocks $h(b_i, k_i)$ comprises bit-wise adding a constant d_i to said output sub-block $h(b_i,k_i)$, said constant d_i ~~substantially~~ having equal length as said sub-block m_i.

28. (Previously presented) A program segment as claimed in claim 27, wherein deriving said sub-blocks t_i from said output sub-blocks $h(b_i, k_i)$ further comprises raising $h(b_i,k_i) \oplus d_i$ to a power 2^i, using said predetermined Galois Field multiplication.

29. (Previously presented) A program segment as claimed in claim 26, wherein deriving said sub-blocks t_i from said output sub-blocks $h(b_i, k_i)$ comprises raising said output sub-block $h(b_i,k_i)$ to a power 2^i, using said predetermined Galois Field (GF) multiplication.

30. (Currently amended) A program segment as claimed in claim 24, wherein said combined data is formed by:

swapping the sub-blocks t_i and t_{2n-1-i}, for i = 0 to n-1~~; and~~

~~concatenating the swapped sub-blocks~~.

31. (Previously presented) A program segment as claimed in claim 26, wherein said sub-block m_i comprises eight data bits, and wherein said multiplying of two elements b and c of $GF(2^8)$ comprises executing a series of multiplications and additions in $GF(2^4)$.

32. (Previously presented) A program segment as claimed in claim 31, wherein said multiplying of said two elements b and c comprises:

representing b as $a_0 + a_1.D$ and c as $a_2 + a_3.D$, where a_0, a_1, a_2 and a_3 are elements of $GF(2^4)$, and where D is an element of $GF(2^8)$ defined as a root of an irreducible polynomial $k(x) = x^2 + x + \beta$ over $GF(2^4)$, where β is an element of $GF(2^4)$; and

calculating $(a_0a_2 + a_1a_3\beta) + (a_1a_2 + a_0a_3 + a_1a_3).D$.

33. (Previously presented) A program segment as claimed in claim 32, wherein β is a root of an irreducible polynomial $h(x) = x^4 + x^3 + x^2 + x + 1$ over GF(2).

34. (Previously presented) A program segment as claimed in claim 26, wherein said sub-block m_i comprises eight data bits, and wherein calculating the inverse of an element b of $GF(2^8)$ comprises performing a series of calculations in $GF(2^4)$.

35. (Previously presented) A program segment as claimed in claim 34, wherein calculating the inverse of said element b comprises:

representing b as $a_0 + a_1.D$, where a_0 and a_1 are elements of $GF(2^4)$, and where D is an element of $GF(2^8)$ defined as a root of an irreducible polynomial $k(x) = x^2 + x + \beta$ over $GF(2^4)$, where β is an element of $GF(2^4)$; and

calculating $(a_0^2 + a_0a_1 + a_1^2\beta)^{-1}((a_0 + a_1) + a_1D)$.

36. (Currently amended) A processor for cryptographically converting a digital input block into a digital output block; said processor comprising:

a first input means for obtaining said digital input block;

a second input means for obtaining a first key K1; and

a cryptographic processing ~~portion~~ means arranged to convert the digital input block into the digital output block by executing a non-linear function g for non-linearly merging said selected part M1 and said first key K1 in one step and producing a data block B1 which non-linearly depends on said selected part M1 and said first key K1, where a selected part of said digital output block is derived from said data block B1.

37. (Currently amended) A processor as claimed in claim 36, wherein said processor comprises a third input means for obtaining a second key K2, and wherein said processor is arranged to:

split said digital input block into said selected part M1 and a second part M2 before performing said merging;

perform a non-linear function g^{-1} to merge said second block M2 with said second key K2 in one step, producing a data block B2 as output; said non-linear function g^{-1} being the inverse of said non-linear function g; and

combine data from data in said data block B1 and in said data block B2; said digital output block being derived from said combined data.

38. (Currently amended) A processor as claimed in claim 36, wherein said merging comprises:

splitting said selected part M1 in a first plurality n of sub-blocks $m_0,..,m_{n-1}$ of ~~substantially~~ equal length;

splitting said first key K1 in said first plurality n of sub-keys $k_0,..,k_{n-1}$, ~~substantially~~ having equal length, the sub-key k_i corresponding to the sub-block m_i, for i = 0 to n-1; and

separately processing each of said sub-blocks m_i by executing for each of said sub-blocks m_i a same non-linear function h for non-linearly merging a sub-block b_i derived from said sub-block m_i with said corresponding sub-key k_i in one, sequentially inseparable step and producing said first plurality of output sub-blocks $h(b_i, k_i)$; and

combining sub-blocks t_i derived from said first plurality of said output sub-blocks $h(b_i, k_i)$ to form said data block B1.

39. (Previously presented) A processor as claimed in claim 38, wherein said function $h(b_i,k_i)$ is defined by:

$h(b_i,k_i) = (b_i.k_i)^{-1}$, if $b_i \neq 0$, $k_i \neq 0$, and $b_i \neq k_i$

$h(b_i,k_i) = (k_i)^{-2}$, if $b_i = 0$

$h(b_i,k_i) = (b_i)^{-2}$, if $k_i = 0$

$h(b_i,k_i) = 0$, if $b_i = k_i$,

where the multiplication and inverse operations are predetermined Galois Field multiplication and inverse operations.

40. (Previously presented) A processor as claimed in claim 39, wherein said sub-block m_i comprises eight data bits, and wherein said multiplying of two elements b and c of $GF(2^8)$ comprises:

representing b as $a_0 + a_1.D$ and c as $a_2 + a_3.D$, where a_0, a_1, a_2 and a_3 are elements of $GF(2^4)$, and where D is an element of $GF(2^8)$ defined as a root of an irreducible polynomial $k(x) = x^2 + x + \beta$ over $GF(2^4)$, where β is an element of $GF(2^4)$; and

calculating $(a_0a_2 + a_1a_3\beta) + (a_1a_2 + a_0a_3 + a_1a_3).D$;

and wherein calculating the inverse of an element b of $GF(2^8)$ comprises calculating $(a_0^2 + a_0a_1 + a_1^2\beta)^{-1}((a_0 + a_1) + a_1D)$.